PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION International Bureau



INTERNATIONAL-APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(F1) I-4		1	
(51) International Patent Classification 5:		(11) International Publication Number:	WO 92/01987
G06F 13/00, 15/20	A1		5 2, 61767
3001 107 00, 15720		(43) International Publication Date:	6 February 1992 (06.02.92)

(21) International Application Number:

PCT/US91/05006

(22) International Filing Date:

16 July 1991 (16.07.91)

(30) Priority data:

554,140

16 July 1990 (16.07.90)

US

(71) Applicant: TEKSTAR SYSTEMS CORPORATION [US/ US]; 2415 Parkview Road, Middleton, WI 53562 (US).

(72) Inventor: ELAM, Daryl, B.; 8503 Fairway Place, Middleton, WI 53562 (US).

(74) Agent: CURFISS, Robert, C.; Andrus, Sceales, Starke & Sawall, 100 East Wisconsin Avenue, Suite 1100, Milwaukee, WI 53202 (US). (81) Designated States: AT (European patent), AU, BE (European patent), BR, CA, CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, KR, LU (European patent), NL (European patent), NO, PL, + SE (European patent), SU.

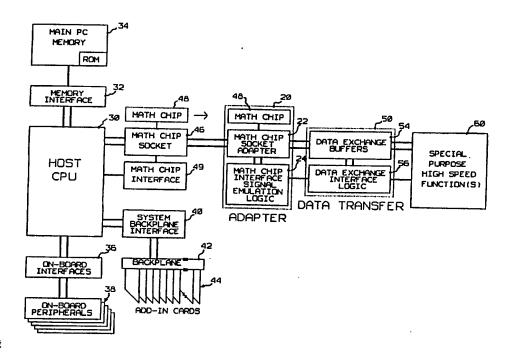
- -

Published

With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: INTERFACE SYSTEM FOR DATA TRANSFER WITH REMOTE PERIPHERAL INDEPENDENTLY OF HOST PROCESSOR BACKPLANE



(57) Abstract

Logic interface circuitry for bypassing the standard backplane bus (40) of a host computer (30) permits the transmission of data between the host computer (30) and special purpose high-speed function add-in peripherals (60). The data is transmitted at the rate it is generated by the host computer (30) and the add-in peripheral (60).

+ DESIGNATIONS OF "SU"

It is not yet known for which States of the former Soviet Union any designation of the Soviet Union has effect.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

• • •					
ΑT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JР	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic	SE	Sweden
CH	Switzerland		of Korea	SN	Senegal
CI	Côte d'Ivoire	KR	Republic of Korea	su+	Soviet Union
CM	Cameroon	LI	Liechtenstein	TD	Chad
		LK	Sri Lanka	TG	Togo
CS	Czechoslovakia	LU	Luxembourg	US	United States of America
DE	Germany	MC	Monaco		
DK	Denmark	WIC.	MUIIACO		

10

15

20

25

30

35

INTERFACE SYSTEM FOR DATA TRANSFER WITH REMOTE PERIPHERAL INDEPENDENTLY OF HOST PROCESSOR BACKPLANE

BACKGROUND OF THE INVENTION

Field of the Invention

The invention relates to a means and method for bypassing standard computer interface backplanes in order to increase the data transfer rate between the host computer and a remote system.

Description of the Prior Art

It has long been recognized that the use of personal computers (PCs) is generally limited by their inability to perform certain functions in a high-speed, efficient manner. This leaves the users of certain special purpose high-speed function add-in peripherals with the choice of adapting to a slow, cumbersome processing operation, or of using a far more expensive mini-computer or mainframe system at many times the cost of a PC whenever efficiency and/or speed is a key factor in the operation.

The backplane bus bandwidth is the most significant factor in restricting the usefulness of high-speed add-in system components which require use of the backplane to communicate with a system central processing unit (CPU). Such components are most efficient when data transfer can be accomplished at the data transfer rate of the CPU rather than at the transfer rate of the backplane. The bandwidth limitations are most pronounced when there is a large discrepancy between the peak data transfer rate capabilities of the system CPU and the backplane bus. Over the years, this discrepancy has become greater and greater, as new generations of new CPUs are introduced far more often than new generations of backplane buses. As CPU capability and speed increase, the backplane becomes the limiting factor in using the CPU with add-in systems to perform more complex tasks.

PCT/US91/05006

5

10 .

15

20

25

30

The primary reason that backplane design lags CPU development is the requirement for standardization. The Institute of Electrical and Electronic Engineers (IEEE) has established standards for backplane architecture which permit numerous manufacturers to design add-in systems for a variety of host CPUs with a high degree of certainty that the mixing and matching of such systems and CPUs will result in an operable system. While this approach assures flexibility, the tradeoff is a lack of efficiency.

In an effort to overcome this, more of the functions benefiting from high speed data transfer are being incorporated onto the host CPU board to avoid using the backplane bus as a communication device. Examples that have become accepted are numerics coprocessor chips, mass storage controllers, video controls and local network interfaces. Physical limitations of the host CPU board render continuing movement in this direction impractical, if not impossible.

The present invention is intended to permit a wide variety of add-in systems to operate at a high data transfer rate by bypassing the backplane of the host processor without the requirement that the add-in system be physically located on the CPU board. The invention takes advantage of the availability of the functional signals connected to and available at sockets generally provided in the standard CPU board. The invention uses the available sockets to permit development of a communications interface to shunt around the backplane bus. An adapter and data processing logic are connected to a selected data access point to provide a data transfer path for connecting the host CPU with the add-in systems. By

WO 92/01987 PCT/US91/05006

5

10

15

20

25

30

3.

using the invention, the data transfer rate can be increased by a factor approaching a magnitude over the capability of the backplane.

SUMMARY OF THE INVENTION

The present invention comprises a means and method for circumventing the limitations of the data communications speed of standard computer interface buses. The backplane bus restricts the usefulness of a certain class of add-in system components, generally in the form of add-in circuit boards, which use the backplane bus to communicate with the system CPU. This class is characterized primarily by the desirability to tranfer data to and/or from the system CPU as quickly and efficiently as possible. The invention utilizes the standardization of the signal connections in the system CPU and/or other peripheral sockets in the CPU board as a convenient location to access the required signals and shunt around the system backplane bus. This may be accomplished by inserting a small socket adapter into the selected socket and then relocating the displaced chip into a similar socket provided on the adapter. In this manner, the protocol established between the displaced chip and the host CPU is maintained and the adapter borrows the data available from the CPU at the socket. Thus, the accessing schemes and handshake protocol of the original, now displaced chip are preserved and the host CPU operates as if it is communicating only with the chip. signals which are available are then lifted from the socket by the adapter for communicating with the remote add-in system.

In an alternative embodiment, the interface includes dedicated interface logic which duplicates the protocol and accessing schemes of the host CPU or the

PCT/US91/05006

5

10

15

20

25

30

displaced chip. This permits direct communication between the host CPU and the adapter circuitry without the use of the displaced chip.

The adapter provides a hard wired system which electrically connects the required signals available at the selected data access point to a remote add-in system to provide a high speed data transfer bus. The only additional electrical connections required other than those available at the selected socket may be the DC power and ground lines which can be taken directly from the standard backplane.

By way of example, the standard PC backplane, known as an Industry Standard Architecture (ISA) backplane, or alternatively, as the IBM PC-AT bus, provides a data path which is 16 parallel bits, the maximum recommended backplane clock speed being 8 megahertz (MHz). The fastest backplane transfers require 4 clock cycles, yielding a peak bandwidth of 4 megabytes per second. Using the present invention with a 33 MHz clock speed requires the same 4 clock cycles per transfer as is standard. The invention yields a data transfer rate of 33 megabytes per second, an over 8-fold improvement. In the example, an Intel 80386 The invention may be adapted microprocessor is used. to interface with most available microprocessors, numeric chips and other generally available sockets or access points, provided the CPU provides at the socket all signals required by the add-in board. The adapters may be stacked or "piggy-backed" on top of one another to permit multiple add-in boards simultaneously utilizing the same socket. The particular function of the add-in system is incidental to the invention as long as the signals required to operate the add-in system are available at the adapter socket.

The various advantages and features of the

WO 92/01987 PCT/US91/05006

5

invention will be readily apparent from the accompanying drawings and description of the preferred embodiment.

5

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a general block diagram of a host processor in combination with a special purpose high-speed function peripheral add-in and an interface unit incorporating the teachings of the present invention.

10

Fig. 2 is a detailed flow chart of the interface device illustrated in Fig. 1.

Fig. 3 is a diagram of a typical (prior art) state machine which may be an integral part of the interface device shown in Figs. 1 and 2.

15

Figs. 4-13 are a schematic diagram of a data transfer interface device made in accordance with the teachings of the present invention.

20

DESCRIPTION OF THE PREFERRED EMBODIMENT

As shown in Fig. 1, the present invention includes an adapter 20 and a data transfer device or circuit 50. In the embodiment illustrated, the CPU of a standard PC such as, by way of example, an IBM PC-AT compatible system having an Intel 80386 microprocessor 30, includes a standard memory interface 32 and main PC memory 34, typical onboard interfaces 36 and typical onboard peripherals 38. A standard system backplane interface 40 is used to provide communication between the CPU 30 and the backplane 42 for providing data transfer to and from the add-in cards 44.

30

25

Typically, the PC will include numerous peripheral sockets such as, by way of example, the extended math chip (EMC) socket 46 adapted for receiving a standard numerics chip 48 such as, by way of example, the Weitek 3167 or the Intel 80387 math

35

10

15

20

25

30

chip. The math chip interface logic circuitry 49 is also provided to control communication between the PC and the math chip.

The subject invention is specifically adapted for connecting the CPU 30 to special purpose high-speed function (SPHSF) add-in systems 60, bypassing the system backplane 42 by utilizing the direct connections to the full speed CPU signals via the math chip socket 46. The math chip socket of the CPU is a convenient place to gain access to the signals needed for the high-speed direct interface. However, the adapter could also be placed directly on the CPU chip or at any other convenient location on the main CPU circuit board where direct access to the required CPU signals can be gained.

As shown in Fig. 1, adapter 20 includes a math chip socket adapter 22 which is connected directly to the math chip socket via a standard plug and socket arrangement which is designed to mate with the math chip socket 46 provided in the PC. The adapter provides a direct communication link between the math chip socket and the data transfer device 50. The adapter can physically be plug compatible with the math chip socket and coupled to ribbon cables or the like which are, in turn, in direct communication with the data transfer device 50. As an option, the math chip 48 may be plug compatible with the adapter. The math chip interface logic 24 may be located on the adapter 20 or may be remote and coupled via the cable.

The protocol and accessing schemes provided in the math chip is preserved in one of three alternative methods when in utilizing the present invention. In the first, the math chip 48 is simply removed from the math chip socket and reinserted in the math chip socket adapter 22 in parallel with the data

WO 92/01987 PCT/US91/05006

5

10

15

20

25

30

35

7.

transfer logic device 50, whereby the PC continues to communicate directly with the math chip and the adapter "picks off" the signals in a passive or non-invasive manner and transmits them to the data transfer circuit 50. In the second method, the math chip interface signal emulation logic circuit 24 emulates the accessing and signal protocol scheme of the math chip to duplicate those of the math chip, whereby the CPU communicates with the adapter as if it were a math chip. In a third method and as described herein, the math chip is only emulated to the degree necessary to permit communication with the CPU. The system has distinct interface characteristics, only emulating as much as necessary of the math chip to permit communication.

The physical configuration of the socket adapter is not critical to the function of the invention except for the requirement that the pin and socket arrangement of the adapter must be compatible with the selected data access point on the CPU board. The adapter may be a structural part of the special purpose circuit board including the math chip interface signal emulation logic 24 and the data transfer logic 50, or it may be remote from the board and connected to it by using cables. Likewise, the SPHSF 60 may be plugged directly into the adapter and/or data transfer board or may be plugged into a socket which then communicates via cable with the data transfer board. The signals to be transmitted from the host CPU 30 to the SPHSF 60 are communicated to the data transfer device 50 via the adapter 20. The signals to be transmitted from the SPHSF 60 to the CPU 30 are communicated to the data transfer device 50 directly from the SPHSF 60. Data exchange buffers 54 receive the data at the rate it is generated and supply it to

10

15

20

25

30

the receiving processor on demand at the rate it can be used. This permits use of asynchronous CPUs and SPHSF The data exchange interface logic device 56 provides interfacing and control logic for controlling the buffer function in response to the CPU 20 and SPHSF 60.

An expanded diagram of the data exchange circuitry 50 is illustrated in Fig. 2. As there shown, three primary communication paths exist between the adapter 20 and the data exchange circuitry 50. first is a host PC data bus 25 which directly links the CPU signals present at the math socket with the exchange circuitry 50. The host PC control bus 26 provides a path for communication of the various control signals between the CPU 30 and data exchange interface logic 56. In addition, PC address bus 27 and "selected" signal line 28 provide for the address and selection signals to be transmitted between the PC and the data exchange circuitry via adapter 20. The data transfer and control signals between the data exchange circuit 50 and the SPHSF 60 are provided at buses 51 and 52 and 33, respectively. The data transfer rate is at the respective processing rate of the SPHSF 60 and the host CPU 30. By using this scheme, the CPU and the SPHSF can transfer data to and from each other at the speed of each of the systems, rather than at the speed permitted by standard backplane architectures.

By way of example, by using the adapter 20 and the data transfer circuitry 50 of the present invention, an over 800% advantage can be gained in data transfer rate (DTR) over that available from a standard IBM PC-AT bus. Using the following formula to determine data transfer rate in megabytes per second (MBs):

WO 92/01987 PCT/US91/05006

 $DTR = \frac{RxWx(1-L)}{Cx8}, \text{ where}$

R = the bus clock rate in megahertz,
W = the width of the data bus in bits,
L = the loss due to refresh,

C = clock periods per transfer, and
8 = the number of bits per byte.

Assuming the fastest uni-directional software instructions are used for data transfer and that these are the "string move" instructions of the Intel 386 CPU for an IBM PC-AT such as, by way of example, "REP MOVSW", and the effects of the PC main memory refresh are equal in both cases, the following calculations can be made:

For the PC-AT Bus:

 $DTR = \frac{RxWx(1-L)}{Cx8}$

 $= \frac{8 \times 16 \times (1 - .03)}{4 \times 8}$

= 3.88 mbs

5

10

15

20

Data Exchange of the Invention:

 $DTR = \frac{RxWx(1-L)}{Cx8}$

5

33x32x(1-.03) 4x8

= 32.01 mbs

10

15

20

25

Thus, for the specific example, the invention provides an increase in data transfer rate of

32.01 3.88 , or 825%

Basically, the data exchange circuit 50 operates as a buffer for storing data produced by the host CPU 30 and the SPHSF 60. The buffers 54 collect information as quickly as it is produced and release it on demand, as needed. The interface logic 56 monitors the SPHSF 60 and the buffers 54 to determine when the buffers are ready to receive data and when the buffers have data available to transfer relative to the SPHSF. The host CPU interface logic 57 communicates with the math chip socket adapter and through it with the host CPU 30 to similary monitor and determine when the buffers are ready to receive and/or transfer data relative to the host CPU.

30

35

With specific reference to the exchange buffers 54, the preferred embodiment employs two FIFO registers 100, 102. The FIFO 100 receives data from the host CPU 30 via the adapter 20 and makes it available to the SPHSF 60. The FIFO 102 receives data from the SPHSF 60 and makes it available to the host

WO 92/01987 PCT/US91/05006

5

10

15

20

25

30

11

CPU 30 via the adapter 20. The dual port RAM 104 which allows both the host CPU 30 via the socket adapter 20 and/or the SPHSF 60 to read and write data autonomously and asynchroncusly in at random addresses. The dual port RAM provides substantial flexibility in the data exchange sequence. The SPHSF, for example, accesses stored data at random as needed, or can "look" at data in the RAM without deleting it. The host can, for example, enter and change commands in the RAM whether or not the SPHSF ever accesses or utilizes the them. The FIFOs are each one-way circuits, as their name implies, and whatever data goes in, comes out in the same order.

If the SPHSF is of the type including a programmable component such as a microprocessor, then the host CPU is adaptable to provide instruction codes to the programmable component via the data exchange circuitry. Specifically, the data exchange circuitry is operable to transfer not only operand data but instruction codes and commands, as well.

In order to make the transfer scheme functional when plugged into the math chip socket, the data exchange circuitry must be able to communicate with the host CPU 30. This can be accomplished by using one of the three following methods:

The numerics chip may be plugged into the adapter, whereby the CPU 30 continues to communicate directly with the numerics chip and the data available at the adapter is lifted from the chip socket for use by the data exchange circuitry and the SPHSF;

(2) The math chip interface signal emulation logic circuitry 24 includes logic for duplicating

10

15

20

25

30

35

the protocol and accessing schemes of the numerics chip to simulate the identity of the numerics chip, permitting the host CPU 30 to function as if it were communicating directly with a standard numerics chip; or

(3) The data exchange circuitry can have a distinct identity provided that specific signals present in the math chip are handled in a manner compatible with the host CPU, i.e., the interface emulates the selected, required minimum protocol of the math chip while ignoring superfluous signals not relevant to the SPHSF.

An example for providing signal emulation is incorporated in the state machine shown in Fig. 3 which is the architecture for a typical state machine for the Intel 80386 microprocessor is illustrated in Fig. 3. The information for creating the state machine is directly available from the Intel 80386 Users Manual. The state machine logic is loaded into PLD circuits provided in the math chip CPU control logic circuitry 108. The math chip CPU control logic circuit communicates directly with the host CPU 30 through the math chip socket adapter 20 and emulates protocol and accessing schemes of a numerics chip, permitting the CPU 30 to function as if it were communicating with the numerics chip.

The math chip CPU control logic 108 communicates directly with the math chip socket to provide emulation. The address decoder is standard decoder architecture similar to that used in the numerics chip or other add-in function peripherals and provides the required handshaking signals between the peripheral and the host CPU 30.

10

15

20

25

30

35

The SPHSF 60 can be any special purpose highspeed functional peripheral. The SPHSF may include a CPU 120 which is adapted for receiving and sending data signals to and from the host CPU 30. In the prior art systems, the SPHSF would have to communicate directly with the host processor through the standard backplane. Where the SPHSF was capable of providing high-speed functions, the system efficiency was entirely dependent upon the speed of the backplane. By using the subject invention, the data exchange circuitry can receive data between the SPHSF 60 and the host CPU 30 at the speed it is generated and supply on demand, as needed. The peak data transfer rate is the data utilization rate of the SPHSF. The SPHSF interface logic circuit 56 in the data exchange circuitry would be customized for each specific The SPHSF may be expandable and adapted to include additional peripherals, as illustrated by DRAM control 122 and DRAM expansion board 124.

It will be readily appreciated that the use of the data transfer scheme of the present invention permits utilization of special purpose high-speed functional add-in peripherals with stand alone computer systems, wherein the rate of data transfer between the SPHSF and the host computer is limited only by the speed at which both the host processor and the add-in peripheral generate and utilize data, rather than being limited by standard backplane architecture. permits the user to buy a relatively inexpensive PC or other microprocessor based host system and add to it the data transfer system of the present invention for substantially less investment than required to purchase a mini-computer or a mainframe, permitting performance levels at mini-computer or mainframe speeds. For example, using the invention with a PC having an Intel

386 microprocessor and an SPHSF includes an Intel 80860 microprocessor, numerical processing sequences can be performed at about half the rate of a CRAY-1 supercomputer at less than 1% of the cost.

5

10

15

A detailed schematic diagram of a data transfer interface device made in accordance with the present invention is illustrated in Figs. 4-13. As illustrated, the cable connectors provide direct access to the host CPU signals and to other useful signals which can be utilized to provide the optimum interface between the host and the add-in SPHSF. In Fig. 4, the upper PLD Ul implements a state machine which tracks the state machine of the host CPU (Fig. 3) by monitoring the signals available via the cable from the adapter. The upper PLD also implements the required handshaking signals according to the host CPU's required protocol for the transfer of data to and from the host. The lower PLD U2 implements the control signals used to interface with the data exchange buffers and other control logic.

20

25

In Fig. 5, the PLD U3 implements the control logic for the exchange of data from the host CPU's side of the data exchange buffers. In this case, they control reading from and writing to FIFO and random access dual port memories. The other side of the FIFOs and dual port memories is written to or read from by the SPHSF. The PAL U4 performs the same functions as the PAL U3 except that this PAL performs it's functions on behalf of the SPHSF, and thus is connected to the logically opposite side of the various buffers and control circuits.

30

In Fig. 6, the four cable connectors P4, P5, P6 and P7 carry the host CPU data bus signals to and from the host CPU socket adapter. In this design there are 32 host data bits, but if there were more or fewer

WO 92/01987 PCT/US91/05006

5

10

15

20

25

30

15

data bits, then the number of buffer logic chips would simply be adjusted to the required number to match the data bus width of the host CPU. The signals on the right side of Fig. 6 are the latched data bus bits from the host CPU or the unlatched data bus bits from the data exchange buffers.

The FIFO circuits in Fig. 7 comprise the lower 32 bits of the SPHSF data bus, which are also the even 32 bit words of the host CPU. These FIFOs pass data unidirectionally from the host to the SPHSF. FIFO circuits in Fig. 8 comprise the upper 32 bits of the SPHSF data bus, which are also the odd 32 bit words of the host CPU. These FIFOs pass data unidirectionally from the host to the SPHSF. circuits in Fig. 9 comprise the lower 32 bits of the SPHSF data bus, which are also the even 32 bit words of the host CPU. These FIFOs pass data unidirectionally from the SPHSF to the host. The FIFO circuits in Fig. 10 comprise the upper 32 bits of the SPHSF data bus, which are also the odd 32 bit words of the host CPU. These FIFOs pass data unidirectionally from the SPHSF to the host.

The dual port RAM chips U25 and U26 in Fig. 11, and U27 and U28 in Fig. 12 are each one megabit memories organized as 64K by 16 bits and are of the fully asynchronous dual ported variety. All signals on the left sides of the chips are connected to the host CPU and/or host interface control circuitry, and vise versa, all signals on the right side are connected to the SPHSF. This provides a random access memory for applications and algorithms not well suited for the sequential nature of FIFOs. The RAMS U25 and U26 from the lower 32 bits and the RAMS U27 and U28 from the upper 32 bits of the dual port RAM 104.

The dual port RAM chips U27 and U28 in Fig.

10

15

20

12 are each one megabit memories organized as 64K by 16 bits and are of the fully asynchronous dual ported variety. All signals on the left sides of the chips are connected to the host CPU and/or host interface control circuitry, and vice versa, all signals on the right side are connected to the SPHSF unit. This provides a random access memory for applications and algorithms not well suited for the sequential nature of FIFOs.

The SPHSF shown in Fig. 13 is shown merely as an example and is a 64 bit high speed RISC processor which is to be used primarily as an accelerator for numerical computation intensive applications. The high speed interface to the host could be utilized for virtually any special function which would benefit from high data transfer rates. Examples would be mass storage controllers, signal processing sub-systems, image processors and the like.

While certain features and embodiments of the invention have been described herein, it will be understood that the invention includes all alternatives encompassed within the scope and spirit of the following claims.

10

15

5

10

CLAIMS

What is claimed is:

- 1. A host/peripheral interface for providing the transfer of data between a host processor and a remote special purpose high-speed system, the transfer means comprising:
- a. an adapter for connecting said interface directly to a data access point associated directly with the host processor;
- b. logic means in communication with the adapter for establishing protocol and access schemes acceptable to the host processor for signaling the host processor to send and receive data at the data access point; and
- c. a data transfer device in direct communication with the adapter and the remote system for transmitting data between the host processor and the remote system at a data transfer rate equal to the rate said data is generated by the respective host processor and/or remote system.
- 2. The host/peripheral interface of Claim 1, the data transfer device further comprising:
- a. a first buffer means in communication with the adapter and the remote system for receiving data generated by the host processor and for storing the data for release to or use by the remote system on an as-needed basis; and
- b. a second buffer means in communication with the remote system and the adapter for receiving data generated by the remote system and for storing the data for release to or use by the host processor; and
- 3. The host/peripheral interface of Claim 2, further including a data exchange logic device in communication with the adapter, the remote system and the buffer means for signaling to each buffer means

5

5

5

- when the remote system is ready to use or receive the data stored in the respective buffer means.
 - 4. The host/peripheral interface of Claim 2, wherein each buffer means further comprises:
 - a. a first-in, first-out data storage
 device; and
 - b. a dual port memory in communication with the adapter and the host processor and the remote system.
 - 5. The host/peripheral interface of Claim 4, each first-in, first-out register having a 512x64 configuration.
 - 6. The host/peripheral interface of Claim 3, the host of the type generating control signals separate and distinct from the data signals, the data exchange logic device further comprising a control logic means for receiving the control signals and transmitting said control signals to the respective buffer means.
 - 7. The host/peripheral interface of Claim 1, the host processor of the type including a standard extended math chip socket, the adapter further including means for connecting said interface device directly into said extended math chip socket.
 - 8. The host/peripheral interface of Claim 7, the extended math chip socket connecting means further including means for connecting a numerics chip in the extended math chip socket in parallel with said host/peripheral interface device.
 - 9. The host/peripheral interface of Claim 7, the logic means further including means for emulating the protocol and accessing schemes of a standard numerics chip.
 - 10. A method for bypassing the backplane of a host processor when transmitting data between the host

processor and a remote special purpose high-speed addin system, comprising the steps of:

- 5 a. selecting a data access point on the host processor;
 - b. emulating the protocol and access schemes required for the host processor to send and receive data to the access point;
- 10 c. collecting and storing data generated and transmitted by the host processor and the remote system; and
 - d. using the stored data.
 - 11. The method according to Claim 10, the host processor of the type including an extended math chip socket, wherein the signals generated and transmitted by the host processor are collected at said socket, and wherein the signals generated and transmitted by the remote system are transmitted to said socket.

BACKPLANE

J 36

ON-BOARD INTERFACES

40

INTERFACE

BACKPLANE

SYSTEM

49

44

ADD-IN CARDS

PERIPHERALS

ON-BOARD

48

94 84

MAIN PC MEMORY 82

Д М

48

46

MATH OFF

NTERFACE

MEMORY

8

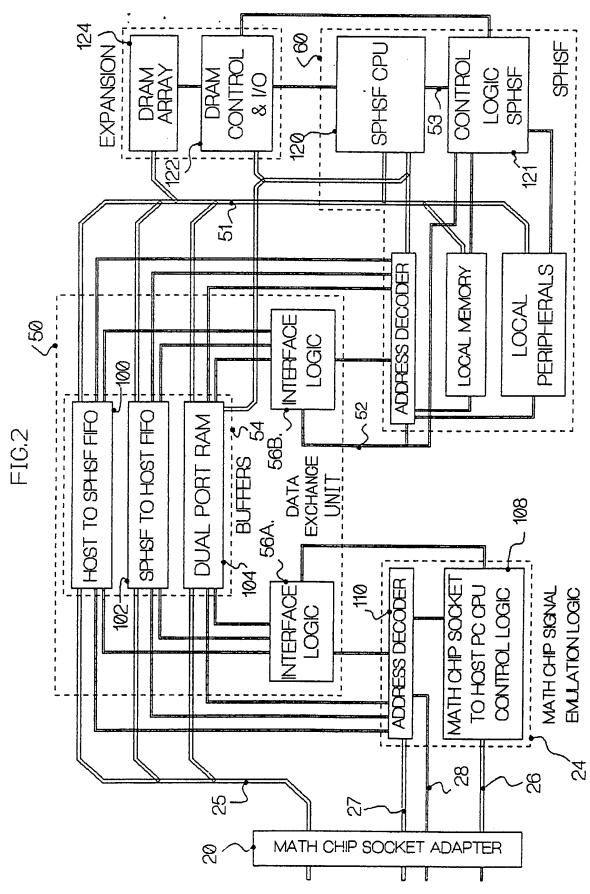
MATH OFF

SOCKET

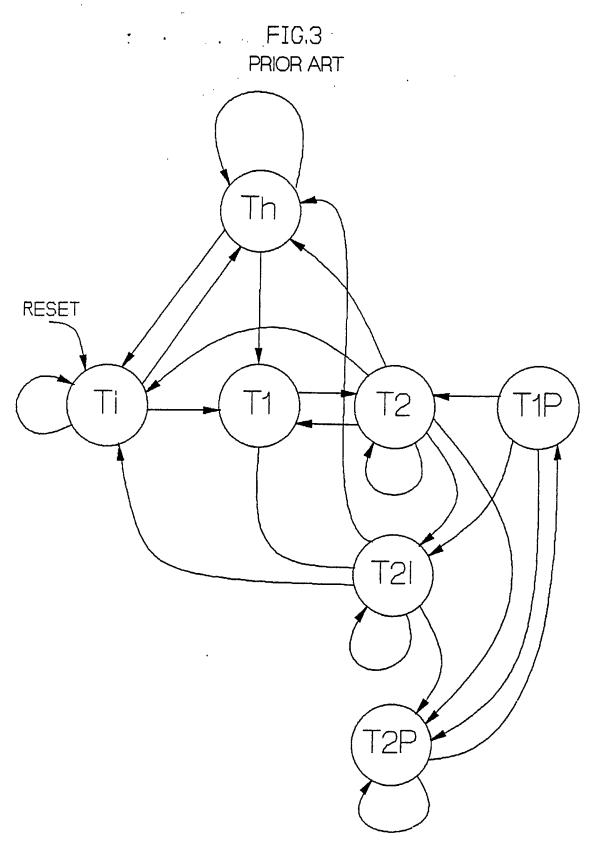
SUBSTITUTE SHEET

MATH CHIP INTERFACE

> HOST OPU

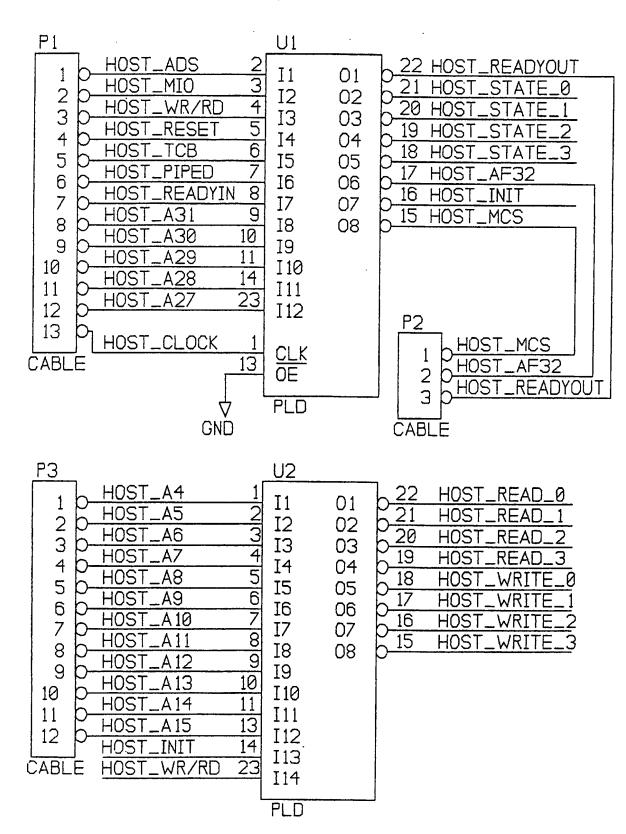


SUBSTITUTE SHEET



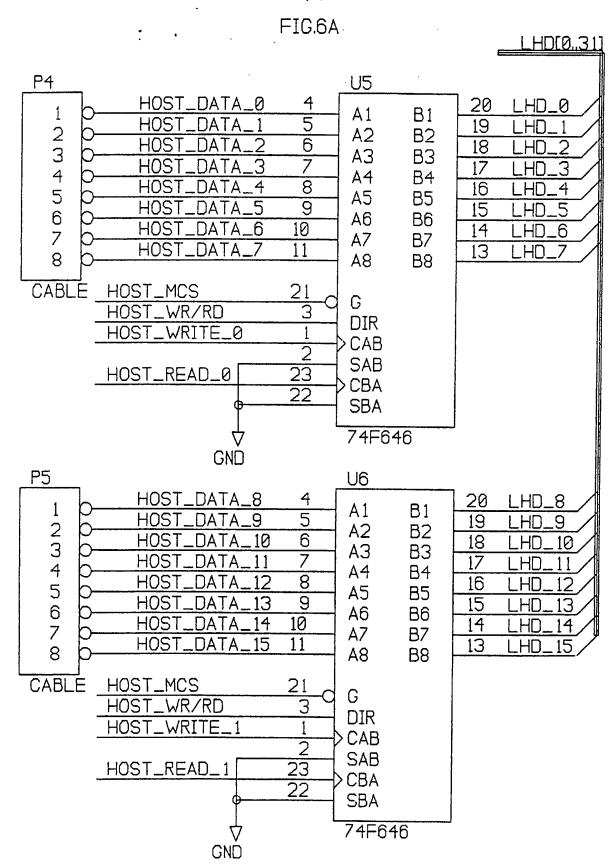
386 DX COMPLETE BUS STATES (INCLUDING PIPELINED ADDRESS)

FIG.4

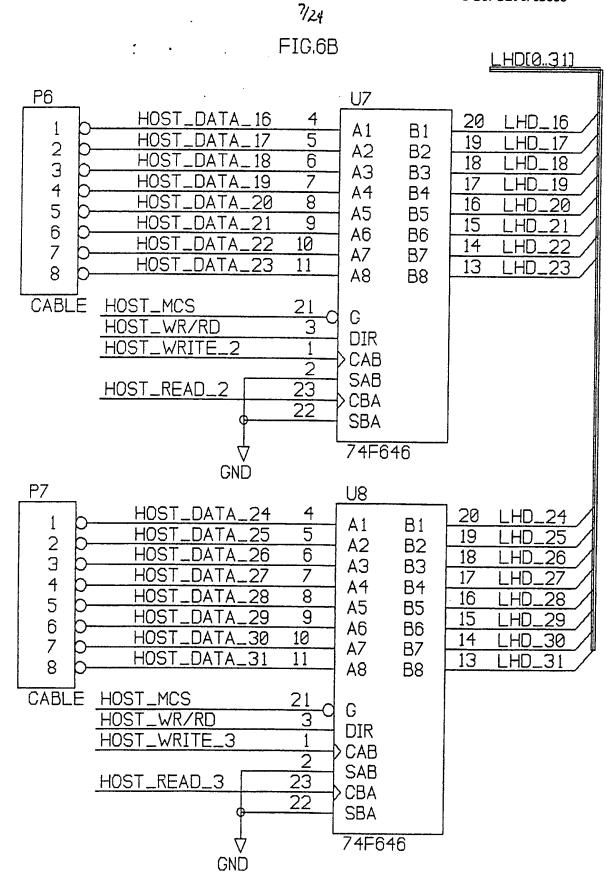


HOST I/O CONTROL LOGIC

WO 92/01987



HOST DATA EXCHANGE REGISTERS



HOST DATA EXCHANGE REGISTERS

8/24 FIG.7A HOST_DATA_[0.,31] SPHSF_DATA_[0,,63] U9 LHD_0 9 SPHSF_DATA_0 DØ Q0 LHD_1 SPHSF_DATA 10 Q1 D1 4 LHD. SPHSF_DATA D2 Q2 LHD 12 SPHSF_ D3 **Q3** LHD 16 SPHSF_ D4 Q4 26 LHD_5 17 SPHSF_ D5 Q5 18 LHD_6 SPHSF_DATA D6 Q6 24 LHD_7 19 SPHSF_DATA D7 Q7 13 **D8 Q8** \overline{XI} XO/HF 21SPHSF_FIFO_EMPTY FL/RT HOST_RESET SPHSF_FIFO_FUL RST 0115 SPHSF_FIFO_READ_ RCLK HOST_FIFO_WRITE_ D WCLK 7201 U10 LHD_8 9 SPHSF_DATA_ D0 Q0 LHD_9 10 SPHSF_DATA D1-Q1 LHD _10 11 SPHSE D2 Q2 3 LHD .11 12 SPHSF_ .DATA D3 Q3 .12 _HD 16 SPHSF D4 Q4 LHD 26 .13 17 SPHSF_ .DATA D5 Q5 25 LHD 14 18 SPHSF_DATA D6 Q6 .HD 19 SPHSF_DATA D7 Q7 2 13 **D8 Q8** 20 XO/HF 21 FL/RT EF HOST_RESET 8 RST SPHSF_FIFO_READ_ _01 RCLK HOST_FIFO_WRITE_0 WCLK

HOST TO SPHSF FIFO BITS 0 TO 15

7201

9/24 FIG.7B HOST_DATA_[0,.31] SPHSF_DATA_[0,63] U11 9 SPHSF_DATA_16 LHD_16 DØ Q0 LHD. _ 17 SPHSF_DATA D1 Q1 LHD_18 SPHSF_DATA D2 Q2 LHD. _ 19 SPHSF_DATA D3 Q3 LHD_20 16 SPHSF_DATA D4 Q4 26 LHD₂1 17 SPHSF_DATA D5 Q5 25 LHD_22 18 SPHSF_DATA D6 Q6 24 LHD₂₃ SPHSF_DATA 19 D7 Q7 13 **D8** Q8 XI XO/HF FL/RT EF 8 HOST_RESET RST SPHSF_FIFO_READ_0115 >RCLK HOST_FIFO_WRITE_0 > WCLK 7201 U12 9 SPHSF_DATA LHD_24 D0 Q0 LHD_25 10 SPHSF_DATA Di Q1 LHD. _26 SPHSF_ .DATA Q2 D2 _27 12 SPHSF_DATA LHD D3 Q3 27 _28 LHD 16 SPHSF_ .DATA D4 Q4 26 SPHSF_DATA LHD_29 17 **D5** Q5 25 LHD_30 18 SPHSF_DATA D6 Q6 LHD_ .31 19 SPHSF_DATA. D7 Q7 13 **D8** Q8 20 XO/HF XI 21 FL/RT EF 8 HOST_RESET RST SPHSF_FIFO_READ_01 RCLK HOST_FIFO_WRITE_ WCLK 7201

HOST TO SPHSF FIFO BITS 16 TO 31

10/24

: · · FIG.8A

HOST_DATA_[031]			SPHSF_DATA_[0.,63]
LHD_0 9 LHD_1 10 LHD_2 11 LHD_3 12 LHD_4 16 LHD_5 17 LHD_6 18 LHD_7 19	U13 Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8	D0 6 D1 4 D2 3 D3 27 D4 26 D5 25 D6 25 D7 2 D8	SPHSF_DATA_32 SPHSF_DATA_33 SPHSF_DATA_34 SPHSF_DATA_35 SPHSF_DATA_36 SPHSF_DATA_37 SPHSF_DATA_38 SPHSF_DATA_38
HOST_FIFO_EMPTY_1 21 HOST_FIFO_FULL_1 8	XO/HF EF FF 7201	PC 15	HOST_RESET HOST_FIFO_READ_1 SPHSF_FIFO_WRITE_01
LHD_8 9 LHD_9 10 LHD_10 11 LHD_11 12 LHD_12 16 LHD_13 17 LHD_14 18 LHD_15 19	Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8	D0 6 D1 4 D2 3 D3 27 D4 26 D5 25 D6 24 D7 2	SPHSF_DATA_40 SPHSF_DATA_41 SPHSF_DATA_42 SPHSF_DATA_43 SPHSF_DATA_44 SPHSF_DATA_45 SPHSF_DATA_45 SPHSF_DATA_46 SPHSF_DATA_47
20 21 8	X0/HF EF FF 7201	RCL K 15	HOST_RESET HOST_FIFO_READ_1 PHSF_FIFO_WRITE_01

SPHSF TO HOST FIFO BITS 32 TO 47

: FIG.8B

HOST_DATA_[0,,31]		SPHSF_DATA_[063]
LHD_16 9 LHD_17 10 LHD_18 11 LHD_19 12 LHD_20 16 LHD_21 17 LHD_22 18 LHD_23 19 13	U15 Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8	D0 D1 D2 D3 D4 D5 D6 D7 D8	6 SPHSF_DATA_48 5 SPHSF_DATA_49 4 SPHSF_DATA_50 3 SPHSF_DATA_51 27 SPHSF_DATA_52 26 SPHSF_DATA_53 25 SPHSF_DATA_54 24 SPHSF_DATA_55 2
20 21 8	7201	XI FL/RT RST RCLK< WCLK<	7 SPHSF_FIFO_WRITE_01 23 HOST_FIFO_READ_1 22 15 HOST_RESET 1
LHD_24 9 LHD_25 10 LHD_26 11 LHD_27 12 LHD_28 16 LHD_29 17 LHD_30 18 LHD_31 19 13	016 01 02 03 04 05 06 07 08	D0 D1 D2 D3 D4 D5 D6 D7	6 SPHSF_DATA_56 5 SPHSF_DATA_57 4 SPHSF_DATA_58 3 SPHSF_DATA_59 27 SPHSF_DATA_60 26 SPHSF_DATA_61 25 SPHSF_DATA_62 24 SPHSF_DATA_63 2
20 21 8	X0/HF EF FF 7201	XI FL/RT RST RCLK< WCLK<	7 SPHSF_FIFO_WRITE_01 23 HOST_FIFO_READ_1 22 15 HOST_RESET 1

SPHSF TO HOST FIFO BITS 48 TO 63

i2/₂₄ : FIG.9A

HOST_DATA_[0,.31]		SPHSF_DATA_[0.63]
LHD_0 9 LHD_1 10 LHD_2 11 LHD_3 12 LHD_4 16 LHD_5 17 LHD_6 18 LHD_7 19 13	U17 Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8	D0 6 SPHSF_DATA_0 D1 5 SPHSF_DATA_1 D2 4 SPHSF_DATA_2 D3 27 SPHSF_DATA_4 D5 26 SPHSF_DATA_5 D6 D7 24 SPHSF_DATA_7 D8 2
HOST_FIFO_EMPTY 21 HOST_FIFO_FULL 8	X0/HF EF FF 7201	TI 7 23 FL/RT 22 HOST_RESET RST 15 HOST_FIFO_READ_0 1 SPHSF_FIFO_WRITE_01
LHD_8 9 LHD_9 10 LHD_10 11 LHD_11 12 LHD_12 16 LHD_13 17 LHD_14 18 LHD_15 19 13	U18 Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8	D0 6 SPHSF_DATA_8 D1 5 SPHSF_DATA_9 4 SPHSF_DATA_10 D2 3 SPHSF_DATA_11 D3 27 SPHSF_DATA_12 D4 26 SPHSF_DATA_13 D5 25 SPHSF_DATA_14 D7 24 SPHSF_DATA_15 D8 2
20 21 8	X0/HF EF FF 7201	TI 7 23 FL/RT RST 22 HOST_RESET 15 HOST_FIFO_READ_0 1 SPHSF_FIFO_WRITE_01

SPHSF TO HOST FIFO LOWER BITS 0 TO 15

: · FIG.9B

HOST_DATA_[0	.311		SPHSE_DATA_[0,63]
LHD_16 9 LHD_17 10 LHD_18 11 LHD_19 12 LHD_20 16 LHD_21 17 LHD_22 18 LHD_23 19 13	U19 Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8	DØ D1 D2 D3 D4 D5 D6 D7 D8	6 SPHSF_DATA_16 5 SPHSF_DATA_17 4 SPHSF_DATA_18 3 SPHSF_DATA_19 27 SPHSF_DATA_20 26 SPHSF_DATA_21 25 SPHSF_DATA_22 24 SPHSF_DATA_23 2
20 21 8	X0/HF EF FF 7201 U20	XI FL/RT RST RCLK< WCLK<	7 23 22 HOST_RESET 15 HOST_FIFO_READ_0 1 SPHSF_FIFO_WRITE_01
LHD_24 9 LHD_25 10 LHD_26 11 LHD_27 12 LHD_28 16 LHD_29 17 LHD_30 18 LHD_31 19 13	QØ Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8	D0 D1 D2 D3 D4 D5 D6 D7 D8	6 SPHSF_DATA_24 5 SPHSF_DATA_25 4 SPHSF_DATA_26 3 SPHSF_DATA_27 27 SPHSF_DATA_28 26 SPHSF_DATA_29 25 SPHSF_DATA_30 24 SPHSF_DATA_31 2
20 21 8	X0/HF EF FF 7201	XI FL/RT RST RCLK< WCLK<	7 23 22 HOST_RESET 15 HOST_FIFO_READ_0 1 SPHSF_FIFO_WRITE_01

SPHSF TO HOST FIFO LOWER BITS 16 TO 31

14/24

: FIG.10A

HOST_DATA_[0,31]		SPHSF_DATA_[0,63]
LHD_0 9 LHD_1 10 LHD_2 11 LHD_3 12 LHD_4 16 LHD_5 17 LHD_6 18 LHD_7 19 13	U21 Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8	D0 6 SPHSF_DATA_32 5 SPHSF_DATA_33 4 SPHSF_DATA_34 D3 3 SPHSF_DATA_35 D4 27 SPHSF_DATA_36 D5 26 SPHSF_DATA_37 D6 25 SPHSF_DATA_38 D7 27 SPHSF_DATA_38 D8 24 SPHSF_DATA_39 D8
HOST_FIFO_EMPTY_1 21 HOST_FIFO_FULL_1 8	X0/HF EF FF 7201	FL/RT 23 FL/RT 22 HOST_RESET RST 15 HOST_FIFO_READ_1 WCLK 1 SPHSF_FIFO_WRITE_01
LHD_8 9 LHD_9 10 LHD_10 11 LHD_11 12 LHD_12 16 LHD_13 17 LHD_14 18 LHD_15 19 13	U22 Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8	D0 6 SPHSF_DATA_40 D1 5 SPHSF_DATA_41 D2 3 SPHSF_DATA_42 D3 27 SPHSF_DATA_43 D4 26 SPHSF_DATA_45 D5 25 SPHSF_DATA_46 D7 D8 2
20 21 8	X0/HF EF FF 7201	TI 7 FL/RT 23 RST 22 HOST_RESET RCLK 1 SPHSF_FIFO_WRITE_01

SPHSF TO HOST FIFO BITS 32 TO 47

: . FIG.10B

HOST_DATA_[0	<u>31</u>]		SPHSF_DATA_[063]
LHD_16 9 LHD_17 10 LHD_18 11 LHD_19 12 LHD_20 16 LHD_21 17 LHD_22 18 LHD_23 19 13	U23 Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8	D0 D1 D2 D3 D4 D5 D6 D7 D8	6 SPHSF_DATA_48 5 SPHSF_DATA_49 4 SPHSF_DATA_50 3 SPHSF_DATA_51 27 SPHSF_DATA_52 26 SPHSF_DATA_53 25 SPHSF_DATA_54 24 SPHSF_DATA_55 2
20 21 8	7201 U24	XI FL/RT RST RCLK < WCLK <	7 23 22 HOST_RESET 15 HOST_FIFO_READ_1 1 SPHSF_FIFO_WRITE_01
LHD_24 9 LHD_25 10 LHD_26 11 LHD_27 12 LHD_28 16 LHD_29 17 LHD_30 18 LHD_31 19 13	QØ Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8	D0 D1 D2 D3 D4 D5 D6 D7	6 SPHSF_DATA_56 5 SPHSF_DATA_57 4 SPHSF_DATA_58 3 SPHSF_DATA_59 27 SPHSF_DATA_60 26 SPHSF_DATA_61 25 SPHSF_DATA_62 24 SPHSF_DATA_63 2
20 21 8	X0/HF EF FF 7201	XI FL/RT RST RCLK & WCLK &	7 23 22 HOST_RESET 15 HOST_FIFO_READ_1 1 SPHSF_FIFO_WRITE_01

SPHSF TO HOST FIFO BITS 48 TO 63

: . FIG.11A

HOST DATA [0.3	211			SPHSF_DATA_[0.63]
HOST_A[0.31]	LIOF			SPHSF_A[0.31]
	<u>U25</u>			¬
	T_A0 AA0		BA0	SPHSF_A0
HOS.			BA1	SPHSF_A1
[]	Γ_A2 AA2 Γ_A3 AA3		BA2	SPHSF_A2
)	T_A4 AA4		BA3 BA4	SPHSF_A3 SPHSF_A4
1 [1]	_A5 AA5		BA5	SPHSF_A5
	A6 AA6		BA6	SPHSF_A6
	_A7 AA7 _A8 AA8		BA <i>7</i> BA8	SPHSF_A7
	_A9 AA9		BA9	SPHSF_A8 SPHSF_A9
	_A10 AA10	6	BA 10	SPHSF_A10
	_A11 AA11	4	BA11	SPHSF_A11
	_A12 AA12 _A13 AA13	K	BA 12 BA 13	SPHSF_A12 SPHSF_A13
) ————————————————————————————————————	_A14 AA14	V	BA 14	SPHSF_A14
HOST	_A15 AA15		BA 15	SPHSF_A15
		X		
HOST_RAM_REA	D_0 ARD	. ,	BRD	SPHSF_RAM_READ_01
HOST_RAM_WRI		1	BWR	SPHAF_RAM_WRITE_01
		1		
LHD.	.0 AD0	6	BD0 BD1	SPHSF_DATA_0
LHD_			BD2	SPHSF_DATA_1 SPHSF_DATA_2
	.3 AD3		BD3	SPHSF_DATA_3
LHD_ LHD_	4 AD4		BD4	SPHSF_DATA_4
	.5 AD5 .6 AD6		BD5 BD6	SPHSF_DATA_5 SPHSF_DATA_6
	7 AD7		BD7	SPHSF_DATA_7
LHD_			BD8	SPHSF_DATA_8
LHD_ LHD_			BD9	SPHSF_DATA_9
	10 AD10 11 AD11		BD10 BD11	SPHSF_DATA_10 SPHSF_DATA_11
LHD_	12 AD12		BD12	SPHSF_DATA_12
LHD_	13 AD13		BD13	SPHSF_DATA_13
LHD_ LHD_	14 AD14 15 AD15		BD14 BD15	SPHSF_DATA_14 SPHSF_DATA_15
<u> </u>	10 VD 13		כוטט	SI HSF_UATA_15
	DUAL_	PORT.	_RAM	
14				10 11

DUAL PORT RAM BITS 0 TO 15

۱*۳_{۵۹}* : FIG.11B

LICOT DATA SO S	_		,,,,,,	
HOST_DATA_[0,31				SPHSF_DATA_[0,63]
_HOST_A[0,.31]			•	SPHSF_A[0,31]
	<u>U26</u>			
HOST_ HOST_	A1 AA1 A2 AA2 A3 AA3 A4 AA4 A5 AA5 A6 AA6 A7 AA7 A8 AA8 A9 AA9 A10 AA10 A11 AA11 A12 AA12 A13 AA13	64 K	BA0 BA1 BA2 BA3 BA4 BA5 BA6 BA9 BA10 BA11 BA13	SPHSF_A0 SPHSF_A1 SPHSF_A2 SPHSF_A3 SPHSF_A4 SPHSF_A5 SPHSF_A6 SPHSF_A7 SPHSF_A8 SPHSF_A8 SPHSF_A9 SPHSF_A10 SPHSF_A11 SPHSF_A12 SPHSF_A13
HOST_	A14 AA14	1	BA 14	SPHSF_A14
HOST_	A 15 AA 15		BA 15	SPHSF_A15
HOST_RAM_READ_ HOST_RAM_WRITE LHD_18 LHD_19 LHD_20 LHD_21 LHD_22 LHD_23 LHD_25 LHD_25 LHD_26 LHD_26 LHD_26 LHD_26 LHD_27 LHD_28 LHD_28 LHD_29 LHD_30 LHD_31	AD0 AD1 AD2 AD3 AD4 AD5 AD6 AD7 AD8 AD9 AD10 AD11 AD11 AD11 AD11 AD11 AD11 AD11	× 1 6	BRD BWR BD0 BD1 BD2 BD3 BD4 BD5 BD6 BD7 BD8 BD10 BD11 BD12 BD13 BD13 BD14 BD15	SPHSF_RAM_READ_01 SPHAF_RAM_WRITE_01 SPHSF_DATA_16 SPHSF_DATA_17 SPHSF_DATA_18 SPHSF_DATA_19 SPHSF_DATA_20 SPHSF_DATA_21 SPHSF_DATA_21 SPHSF_DATA_22 SPHSF_DATA_23 SPHSF_DATA_25 SPHSF_DATA_25 SPHSF_DATA_26 SPHSF_DATA_27 SPHSF_DATA_28 SPHSF_DATA_29 SPHSF_DATA_30 SPHSF_DATA_31
	L DUAL_F	PORT.	_RAM	

DUAL PORT RAM BITS 16 TO 31

18/24

	FIG.	12A	
HOST_DATA_[0,31]			SPHSF_DATA_[0,.63]
H0ST_A[031]	U27		SPHSF_A[0,3]]
HOST_A0 HOST_A1 HOST_A2 HOST_A3 HOST_A4 HOST_A5 HOST_A5 HOST_A6 HOST_A7 HOST_A8 HOST_A9 HOST_A9 HOST_A10 HOST_A11 HOST_A12 HOST_A13 HOST_A13 HOST_A15	AA0 AA1 AA2 AA3 AA4 AA5 AA6 AA7 AA8 AA10 AA11 AA12 AA13 AA14 AA15	BA1 SPH BA2 SPH BA3 SPH BA4 SPH BA5 SPH BA6 SPH BA6 SPH BA8 SPH BA9 SPH BA10 SPH BA11 SPH BA12 SPH BA13 SPH BA14 SPH	HSF_A0 HSF_A1 HSF_A2 HSF_A3 HSF_A4 HSF_A5 HSF_A6 HSF_A7 HSF_A8 HSF_A9 HSF_A10 HSF_A11 HSF_A12 HSF_A12 HSF_A13 HSF_A14 HSF_A15
HOST_RAM_READ_1 HOST_RAM_WRITE_1 LHD_0 LHD_1 LHD_2 LHD_3 LHD_4 LHD_5 LHD_5 LHD_6 LHD_7 LHD_8 LHD_9 LHD_10 LHD_11 LHD_12 LHD_12 LHD_13 LHD_13 LHD_13 LHD_13 LHD_13 LHD_14 LHD_15	ARD AWR AD0 AD1 AD2 AD3 AD4 AD5 AD6 AD7 AD8 AD9 AD10 AD11 AD12 AD13 AD14 AD15	BWR SPHED1 SPHED2 SPHED3 SPHED5 SPHED5 SPHED5 SPHEDD9 SPHED11 SPHED11 SPHED12 SPHED13 SPHED13 SPHED14 SPHED1 SPHED1 SPHED14 SPHED14 SPHED14 SPHED14 SPHED14 SPHED14 SP	ISF_RAM_READ_01 IAF_RAM_WRITE_01 ISF_DATA_32 ISF_DATA_33 ISF_DATA_35 ISF_DATA_36 ISF_DATA_36 ISF_DATA_38 ISF_DATA_38 ISF_DATA_39 ISF_DATA_40 ISF_DATA_40 ISF_DATA_41 ISF_DATA_41 ISF_DATA_45 ISF_DATA_45 ISF_DATA_45 ISF_DATA_45 ISF_DATA_46 ISF_DATA_47
	DUAL_POR	T_RAM	

DUAL PORT RAM BITS 32 TO 47

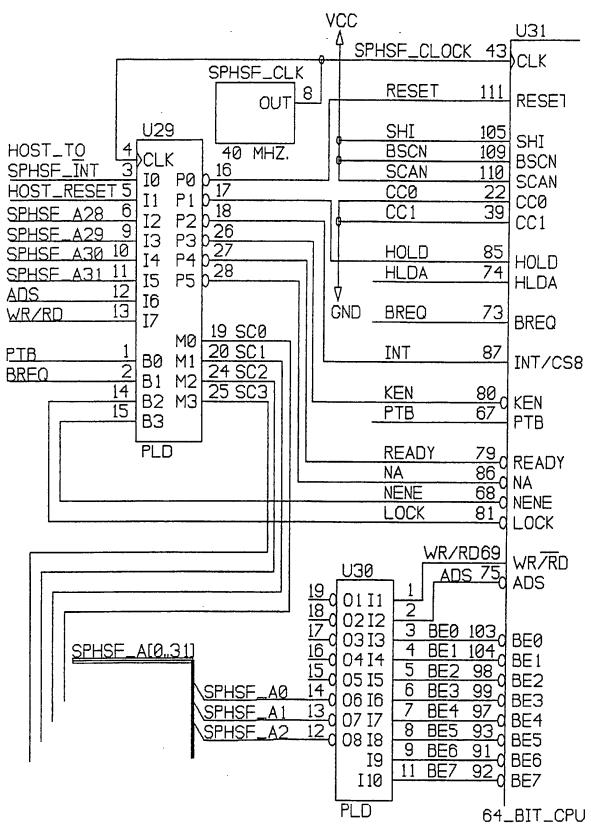
19/24

: · FIG.12B

HOST_A[0,31] HOST_A[0,31] U28 HOST_A0 AA0 BA0 SPHSF_A0 HOST_A1 AA1 BA1 SPHSF_A1 HOST_A2 AA2 BA2 SPHSF_A2 HOST_A3 AA3 BA3 SPHSF_A3 HOST_A4 AA4 BA4 SPHSF_A4 HOST_A5 AA5 BA5 SPHSF_A5 HOST_A6 AA6 BA6 SPHSF_A5 HOST_A7 AA7 BA7 SPHSF_A7 HOST_A8 AA8 BA8 SPHSF_A8 HOST_A9 AA9 HOST_A10 AA10 BA10 SPHSF_A10 HOST_A11 AA11 AB11 SPHSF_A11	
H0ST_A0	
H0ST_A0	
HOST_A1	- 11
HOST_A11 AA11 BA11 SPHSF_A11 BA12 SPHSF_A12 BA13 SPHSF_A12 BA13 SPHSF_A13 BA14 SPHSF_A14 BA14 SPHSF_A15 BA15 SPHSF_A	
HOST_A15 AA15 BA15 SPHSF_A15	
· ×	
HOST_RAM_READ_1	
LHD_24 AD8 BD8 SPHSF_DATA_56 LHD_25 AD9 BD9 SPHSF_DATA_57 LHD_26 AD10 BD10 SPHSF_DATA_58 LHD_27 AD11 BD11 SPHSF_DATA_59 LHD_28 AD12 BD12 SPHSF_DATA_60 LHD_29 AD13 BD13 SPHSF_DATA_61 LHD_30 AD14 BD14 SPHSF_DATA_62 LHD_31 AD15 BD15 SPHSF_DATA_63	
DUAL_PORT_RAM	

DUAL PORT RAM BITS 48 TO 63

عواريو FIG.13A



SPECIAL PURPOSE HIGH SPEED FUNCTION

54/₂₄ : FIG.13B

116 SPHSF_DATA_1 115 SPHSF_DATA_1 132 SPHSF_DATA_2 131 SPHSF_DATA_2 131 SPHSF_DATA_3 147 SPHSF_DATA_4 130 SPHSF_DATA_5 163 SPHSF_DATA_5 163 SPHSF_DATA_6 129 SPHSF_DATA_7 145 SPHSF_DATA_8 146 SPHSF_DATA_9 162 SPHSF_DATA_10 101 101 101 101 101 101 101	U31.		SPHSF_DATA_[063]
146 SPHSF_DATA_9 162 SPHSF_DATA_10 128 SPHSF_DATA_11 161 SPHSF_DATA_12 127 SPHSF_DATA_13 143 SPHSF_DATA_14 144 SPHSF_DATA_15 160 SPHSF_DATA_16 126 SPHSF_DATA_16 126 SPHSF_DATA_17 159 SPHSF_DATA_18 142 SPHSF_DATA_18 142 SPHSF_DATA_19 158 SPHSF_DATA_19 158 SPHSF_DATA_20 125 SPHSF_DATA_21 141 SPHSF_DATA_21 141 SPHSF_DATA_22 124 SPHSF_DATA_23 157 SPHSF_DATA_23 157 SPHSF_DATA_23 158 SPHSF_DATA_25 156 SPHSF_DATA_25 156 SPHSF_DATA_26 123 SPHSF_DATA_26 123 SPHSF_DATA_27 139 SPHSF_DATA_28 122 SPHSF_DATA_29 138 SPHSF_DATA_29 138 SPHSF_DATA_29		D1 D2 D3 D4 D5 D6	115 SPHSF_DATA_1 132 SPHSF_DATA_2 131 SPHSF_DATA_3 147 SPHSF_DATA_4 130 SPHSF_DATA_5 163 SPHSF_DATA_6
126 SPHSF_DATA_17 159 SPHSF_DATA_18 142 SPHSF_DATA_19 158 SPHSF_DATA_20 125 SPHSF_DATA_21 141 SPHSF_DATA_21 141 SPHSF_DATA_22 124 SPHSF_DATA_23 157 SPHSF_DATA_23 158 SPHSF_DATA_25 156 SPHSF_DATA_25 156 SPHSF_DATA_26 123 SPHSF_DATA_27 139 SPHSF_DATA_28 122 SPHSF_DATA_29 138 SPHSF_DATA_29		D9 D10 D11 D12 D13 D14	146 SPHSF_DATA_9 162 SPHSF_DATA_10 128 SPHSF_DATA_11 161 SPHSF_DATA_12 127 SPHSF_DATA_13 143 SPHSF_DATA_14
140 SPHSF_DATA_25 D25		D17 D18 D19 D20 D21 D22	126 SPHSF_DATA_17 159 SPHSF_DATA_18 142 SPHSF_DATA_19 158 SPHSF_DATA_20 125 SPHSF_DATA_21 141 SPHSF_DATA_22
D30 D31 121 SPHSF_DATA_31 SPECIAL PURPOSE HIGH SPEED FUNCTION	·	D25 D26 D27 D28 D29 D30 D31	140 SPHSF_DATA_25 156 SPHSF_DATA_26 123 SPHSF_DATA_27 139 SPHSF_DATA_28 122 SPHSF_DATA_29 138 SPHSF_DATA_30 121 SPHSF_DATA_31

SPECIAL PURPOSE HIGH SPEED FUNCTION

22/24

: FIG.13C

•	. •	SPHSF_DATA_[0.63]
U31		
	D32 D33 D34 D35 D36 D37 D38 D39	120 SPHSF_DATA_32 114 SPHSF_DATA_33 107 SPHSF_DATA_34 108 SPHSF_DATA_35 101 SPHSF_DATA_36 102 SPHSF_DATA_37 100 SPHSF_DATA_38 96 SPHSF_DATA_39
	D40 D41 D42 D43 D44 D45 D46 D47	94 SPHSF_DATA_40 95 SPHSF_DATA_41 89 SPHSF_DATA_42 90 SPHSF_DATA_43 88 SPHSF_DATA_44 83 SPHSF_DATA_45 82 SPHSF_DATA_46 84 SPHSF_DATA_47
	D48 D49 D50 D51 D52 D53 D54 D55	76 SPHSF_DATA_48 77 SPHSF_DATA_49 70 SPHSF_DATA_50 78 SPHSF_DATA_51 64 SPHSF_DATA_52 71 SPHSF_DATA_53 65 SPHSF_DATA_54 72 SPHSF_DATA_55
64_BIT_CPU	D56 D57 D58 D59 D60 D61 D62 D63	58 SPHSF_DATA_56 66 SPHSF_DATA_57 59 SPHSF_DATA_58 53 SPHSF_DATA_59 37 SPHSF_DATA_60 60 SPHSF_DATA_61 38 SPHSF_DATA_62 54 SPHSF_DATA_63

SPECIAL PURPOSE HIGH SPEED FUNCTION

23/24

: FIG.13D

SPHSF_A[031]	
	U31
SPHSF_A3 61	40
SPHSF_A4 62	A3
SPHSF_A5 55	1 A4
SPHSF_A6 49	45
SPHSF_A7 48	-,A6
SPHSF_A8 31	∱ A <i>7</i>
SPHSF_A9 47	48
SPHSF_A10 30	A9
SPHSF_A11 46	A 10
SPHSF_A12 13	A11
	A12
	A 13
SPHSF_A14 45	A14
SPHSF_A15 28	A 15
SPHSF_A16 44	A 16
SPHSF_A17 12	A 17
SPHSF_A18 27	A 18
SPHSF_A19 11	A 19
SPHSF_A20 26	A20
SPHSF_A21 10	A20 A21
SPHSF_A22 42	
SPHSF_A23 9	A22
SPHSF_A24 25	A23
SPHSF_A25 8	A24
SPHSF_A26 41	A25
SPHSF_A27 24	A26
SPHSF_A28 23	A27
SPHSF_A29 7	A28
SPHSF_A30 40	A29
SPHSF_A31 6	A30
<u> </u>	A31
•	64_BIT_CPU

SPECIAL PURPOSE HIGH SPEED FUNCTION SUBSTITUTE SHEET

INTERNATIONAL SEARCH REPORT

International Application No PCT/US91/05006

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) 3						
According to International Patent Classification (IPC) or to both National Classification and IPC IPC (5): GO6F 13/00, 15/20 U.S. C1. 395/325						
	DS SEARCHED	0,0,023				
	Minimum Docum	mentation Searched 4				
Classific						
		Classification Symbols	· · · · · · · · · · · · · · · · · · ·			
U.S.	C1. 395/325,800					
	Documentation Searched othe to the Extent that such Documer	er than Minimum Documentation hts are included in the Fields Searched 6				
	UMENTS CONSIDERED TO BE RELEVANT					
Category	Citation of Document, 1-5 with indication, where a	ppropriate, of the relevant passages 17	Relevant to Claim No. 18			
Y	US, A, 4,246,637 (BROWN) 20 JAI Note (Fig. 1, element 22; Col. Col. 3, lines 36-38)	NUARY 1981 (20.01.81) 2, lines 17-20;	1-11			
Y	US, A, 4,860,244 (BRUCKERT) 22 Note (Abstract; Col. 2 and 3; 1	2–6				
Y,P	US, A, 4,953,930 (RAMSEY) 04 SI Note (Col. 3, lines 18-36; Col. Abstract)	7-9,11				
A	US, A, 4,591,973 (FERRIS, III)	1-11				
A	US, A, 4,309,754 (DINWIDDIE, JR (05.01.82)	1–11				
	·					
•						
* Special categories of cited documents: 15 "A" document defining the general state of the art which is not considered to be of particular relevance "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the						
"E" earl	ier document but published on or after the international of data	invention "X" document of particular relevance				
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)		involve an inventive step	innot be considered to			
"O" doc	tion or other apecial reason (as specified) ument referring to an oral disclosure, use, exhibition or or means	Y" document of particular relevance; cannot be considered to involve an document is combined with one or	inventive step when the			
"P" doc	or means ument published prior to the international filing date but r than the priority date claimed	ments, such combination being ob- in the art. "4" document member of the same pat	rious to a person skilled			
	FICATION					
Date of the Actual Completion of the International Search 2		Date of Mailing of this International Search	th Report ³			
22 OCTOBER 1991 (22.10.91)		11 DEC 1991				
Iternation	el Searching Authority 1	Signature of Authorized Oficer 19 4 on	Ho Nguesen			
ISA/US	S	INTERNATIONAL DIVISION GOPAL C. RAY	! " -			

Form PCT/ISA/210 (second sheet) (May 1986)